

AD-A098 982

CALIFORNIA UNIV LOS ANGELES GRADUATE SCHOOL OF MANAGEMENT F/6 9/2
ISSUES IN SOFTWARE MAINTENANCE AND MEASUREMENT.(U)
MAY 81 B P LIENTZ

N00014-75-C-0266

NL

UNCLASSIFIED

1 1
2 1
3 1
4 1
5 1
6 1
7 1
8 1
9 1
10 1
11 1
12 1
13 1
14 1
15 1
16 1
17 1
18 1
19 1
20 1
21 1
22 1
23 1
24 1
25 1
26 1
27 1
28 1
29 1
30 1
31 1
32 1
33 1
34 1
35 1
36 1
37 1
38 1
39 1
40 1
41 1
42 1
43 1
44 1
45 1
46 1
47 1
48 1
49 1
50 1
51 1
52 1
53 1
54 1
55 1
56 1
57 1
58 1
59 1
60 1
61 1
62 1
63 1
64 1
65 1
66 1
67 1
68 1
69 1
70 1
71 1
72 1
73 1
74 1
75 1
76 1
77 1
78 1
79 1
80 1
81 1
82 1
83 1
84 1
85 1
86 1
87 1
88 1
89 1
90 1
91 1
92 1
93 1
94 1
95 1
96 1
97 1
98 1
99 1
100 1

END
DATE
FILMED
6 81
DTIC

LEVEL II

(12)

unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
	AD A098982	
4. TITLE (and Subtitle)		5. TYPE OF REPORT & PERIOD COVERED
ISSUES IN SOFTWARE MAINTENANCE AND MEASUREMENT		Technical
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s)		8. CONTRACT OR GRANT NUMBER(s)
Bennet P. Lientz		NO0014-75-C-0266
9. PERFORMING ORGANIZATION NAME AND ADDRESS		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
Graduate School of Management University of California, Los Angeles		NR 049-345
11. CONTROLLING OFFICE NAME AND ADDRESS		12. REPORT DATE
Information Systems Program Office of Naval Research, Arlington, Va.		May, 1981
		13. NUMBER OF PAGES
		21
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report)
---		unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report)		
distribution of this document is unlimited		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		

18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)		
Software Maintenance, Information Systems Planning, Enhancements, Measurements		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)		
<p>Up to a few years ago the area of software maintenance was largely ignored. Interest has increased in the last few years due to several factors. First, the increased burden of maintenance from that of ten years ago has restricted resources available for new development. Second, there has been a growing awareness that considering tools which assist development may have little effect on operational systems.</p>		

AD A098982

DTIC
ELECTE
MAY 15 1981

E

DTIC FILE COPY

DD FORM 1473
1 JAN 73

EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

ISSUES IN SOFTWARE MAINTENANCE AND MEASUREMENT*

Bennet P. Lientz
Graduate School of Management
University of California, Los Angeles

Accession For		
NTIS GRA&I	<input checked="checked" type="checkbox"/>	
DTIC TAB	<input type="checkbox"/>	
Unannounced	<input type="checkbox"/>	
Justification		
Distribution/		
Availability Codes		
Avail and/or		
Dist	Special	
A		

*This work was partially supported by the Information Systems Program, Office of Naval Research under contract N00014-75-C-0266, project number NR 049-345.

81 5 14 032

ABSTRACT

Up to a few years ago the area of software maintenance was largely ignored. Interest has increased in the last few years due to several factors. First, the increased burden of maintenance from that of ten years ago has restricted resources available for new development. Second, there has been a growing awareness that considering tools which assist development may have little effect on operational systems. Third, the management of information systems has come under increasing scrutiny.

In this paper we highlight some of the major issues that surfaced during several extensive operational software studies. From this base we can begin to delineate the framework for a future operational software environment.

1. INTRODUCTION

During the past four years an effort has been made to develop a better understanding of software maintenance and enhancement in particular and operational software in general. Several factors motivated this attention. First, it has been widely observed that software maintenance and operational support consume substantial resources in the information systems environment. Although personnel consumption is the most frequently emphasized, hardware and system software are also consumed. Multiple versions of data communications monitors and operating systems are often needed to keep older systems running. This is due to the inability of the application software to migrate to newer releases of systems software.

A second factor is the resource issue in general. Personnel availability is limited. Turnover of systems staff is a major concern for many organizations. In the area of operational software turnover of maintenance personnel can result in reduced support of the application system and even result in damage as untrained or unfamiliar staff attempt to grapple with a particular enhancement or maintenance fix.

A third factor is the sense that much of the software engineering and computer science research has not touched on the problems associated with maintenance and operations. Research has produced many development tools and techniques. Some of these have substantial merit. However, these tools are not easily transferred into a maintenance environment involving large scale operational applications which are over five years old.

An applied research program was initiated to determine what studies and analyses had been carried out. A literature search proved somewhat disheartening. With few exceptions the meagre literature was based on extremely small sample sizes. From such limited data very substantial conclusions were drawn. Some of these in retrospect are worth reviewing. There is the hypothesis that maintenance burdens continue to grow unabated. There is the feeling that staff morale and motivation in maintenance are very low. A third hypothesis was that development tools could be used to reduce maintenance costs. Overall the hypotheses centered on the technical aspects of maintenance. The reader is referred to papers by Belady and Lehman [1], Boehm [2], Boehm et al [3], and Canning [4] for some of the more interesting findings.

The research program began with a small scale survey of less than one hundred systems. The results were reported in Lientz, Swanson, and Tompkins [5]. This survey was based on a fifteen page questionnaire mailed to firms in the western United States. Several interesting results emerged from the survey. First, maintenance and enhancement were found to consume approximately half of the systems and programming personnel hours. Approximately 60% of the effort was in the area of perfective maintenance (i.e., system enhancements, improved documentation, and recoding for efficiency). This finding was somewhat unexpected since the literature had supported the belief that fixing programs and keeping systems operational were the major concerns. A third finding was that problems of a managerial nature were viewed as more significant than those of a technical type.

Interest in maintenance was also increased as a result of this study. Individual contacted during the survey continued to pursue analysis. Several organizations adopted changes suggested as a result of the research. These factors and the small sample size encouraged a larger sample size survey.

The larger survey consisted of a sample of two thousand management members of the Data Processing Management Association (DPMA). This organization was selected because it has the largest percentage of membership based in systems personnel in industrial systems positions. The survey methodology and results are contained in the book Software Maintenance Management (Lientz and Swanson [7]). Summary results and other findings have been presented in Lientz and Swanson [5] and [6]. For background it is useful to highlight the methodology employed in this larger sample.

The DPMA Foundation provided a randomly generated subset of the ten thousand members who classify their jobs in management. The questionnaire was accompanied by an endorsement by the DPMA Foundation. Return envelopes and followup postcards were used to encourage response. There were 486 valid responses. This is quite remarkable considering that the questionnaire was lengthy (over 17 pages) and conducted by mail. The data was entered into a computer and analyzed using the statistical routines in SPSS. Some of the major issues that surfaced are discussed in the next section.

These surveys were conducted with a base consisting mainly of business systems as opposed to real time, sensor based systems. With factory automation, improvements in command and control, and increased on-line systems it was felt that the methodology should be applied to this group of systems.

In 1980 a limited study was undertaken for the Office of the Assistant Secretary of the Navy of eighteen weapon systems. Each software weapon system is in operational use for a particular Navy airplane, missile, or ship class. Most of these systems were real time, fed by sensors and/or radar. The results of this study confirmed the findings of the larger previous study. The Naval weapon study work was conducted with Peter Wegner and will be reported separately.

In the next section we summarize the issues and problems that have been discovered in the areas of application software maintenance and operation. Section 3 presents a possible framework or an approach and suggests specific areas where further work is needed.

2. ISSUES AND PROBLEMS IN APPLICATION SOFTWARE MAINTENANCE

In the surveys four areas of issues have emerged as dominant and comprehensive. We will consider each of these in terms of research as well as implementation concerns.

o Conceptual Issues

At the heart of maintenance is its very definition. In the surveys an inclusive definition was employed. Such a definition includes enhancements and operational hand holding support as part of maintenance along with routine debugging and problem identification and resolution. There are psychological impacts based on a possible derogatory implication of the word maintenance. However, the inclusionary definition helps to aggregate the support needed for an operational application system. The research has shown in all three studies that enhancements for users are the major activity. Adaptation to new technology surfaced only in the weapons systems survey. Emergency fixes and recoding for efficiency were also relatively minor in resource utilization.

Associated with the definition of maintenance is the extensive continued development of an application system. For many systems there appears to be no single life cycle. Rather the life cycle appears to repeat itself. The data appeared to support the view that once development was complete and the system stabilized in operational use, enhancements began individually or in groups. However, the data was not sufficient to fully support this hypothesis. If the hypothesis holds for a particular system, then as users request new enhancements, a new developmental cycle is begun.

What is needed in the conceptual framework of maintenance is a complete classification of the tasks and work done under the maintenance umbrella. Swanson 9 has begun this work. It needs to be further refined. While the concept of maintenance appears academic, there are substantial practical implications as well. A classification method could be used to assist project control systems. The classification into perfective, adaptive, and corrective maintenance is now in use in a number of organizations and has proved beneficial in cost estimation by task and type of system. Systems groups increasingly are charging back their costs to user organizations. A necessary part of the foundation is fairly accurate estimation of costs. The data and classification method assist in this task.

o Measurement of Application Systems

Beyond maintenance is the issue of how to measure a system. Software engineering has concentrated on counting and measuring physical attributes of a system. But size does not tell the entire story. The surveys indicated that systems with very similar sizes revealed entirely different patterns of maintenance activity. The findings of the surveys shed light on an expanded measurement approach. The findings revealed the key role of the user and manager in maintenance activities. This suggests that measurement of software should be done externally as well as internally.

To explore external change sources it is useful to consider the environment of an application system. There are four basic parts of the environment which can affect a system.

- User external

This environment includes legislation, competitive pressures, social and cultural factors. It also includes the internal user organization and staffing. There are quantitative factors here which can be gathered. The requests for change can be classified as to their ultimate source. The number of users actively working with the system can be measured.

- Technological

Technological change can affect applications. Distributed data processing can result in the split of an application across multiple computer systems. New, more intelligent terminals may have the same impact. Technology may also make it possible to join or tie together separate applications.

- Managerial

Management pressure is frequently exerted to control costs and to modify schedules. This pressure can directly impact the maintenance effort and its quality. It is one reason why documentation of changes is frequently not done or is insufficient. Managerial pressure also focuses on the short term. There is a lack of attention to fundamental rework of application systems using new techniques. Who wants to expend the effort to rework something that works? This in turn prevents the use of productivity aids. System size gets larger as enhancement piles upon enhancement. The surveys reveal, not unexpectedly, that systems become more complex and difficult to maintain as they age. They grow in size and complexity. The original staff that know the application attrition out of the organization.

- Marketplace

The marketplace produces new products and services as we have noted in technology. It also creates a competition for personnel, exerting more pressure on the maintenance staff. Furthermore, new products and services may spur the users to request more enhancements.

These four factors yield data which can assist in the measurement process. Yet there has been little attention to externalities in the measurement literature.

o Scale of Effort

The contention in the past has been that the percentage in maintenance is steadily increasing. The surveys do not bear this out. The data indicates that the percentage is relatively stable in most organizations- about 50% of the effort. However, there are organizations in the samples which report sharply rising percentages over a two year period. The respondents in several instances indicated that controls are exerted by management to reduce the percentage. Thus, it appears that scale of effort is heavily dependent on the organizational environment and the portfolio of application systems being developed and maintained at a given time.

o Organizational Issues and the Role of the Users

In the past interest has centered on the organization of maintenance within a systems group. Questions that arise are whether it should be separated or combined with development. However, given the rising interest and impact of the user community it might be well to consider more global issues. What is the role of the users in maintenance and enhancement? Should users be given report generators and other aids? Should users be responsible for production? This is true today in a number of minicomputer based on-line systems.

The role of users is a major issue for systems groups in general. Nationally there is a shortage of 25- 30% in systems personnel. Users may have a role in filling the gap between supply and demand. This is happening today in many organizations and likely to continue as delays lengthen due to staff shortages. Thus, the user role in maintenance, enhancement, and operations needs to be assessed in general.

A separate, but related organizational issue is that of controls for the system. The surveys in the commercial sector reveal that many controls that are supported in education and theory are not used in practice. The issue here is the trade-off between the benefits of the controls and the cost of their control and implementation. Also, many organizations lack the technical implementation aids that make such controls bearable. The issue here is to determine which groups of controls are appropriate to each category of application systems.

o Productivity Issues

A main research focus has been the productivity of programmers and to a lesser extent analysts in the systems organization. A variety of techniques have been devised. The surveys reveal only limited use. Furthermore, in cases where they are employed the results from the survey are not significantly different from traditional methods. It should be emphasized that there was no control or verification of the techniques among the respondents.

But is the productivity of programmers the major concern? The findings cited in the survey results and what we already discussed point to the user and manager areas. There are far more users than developers or maintainers. Thus, if a productivity technique can be found for a user function, its effect is multiplied far more than that for programmers. This also relates to the role of the user that was

discussed earlier. Productivity of users which are performing less complex tasks may be easier to achieve than aiding a programmer with a complex task.

A second area of productivity tools has focused on the analysis and design stages of system development and enhancement. These tools aim at improving the design correctness and completeness. The thought here is that by nailing down the requirements, the system will be easier to maintain and will more completely meet the user needs. This view of the world was probably valid at a time when systems were batch oriented and when users were not involved with systems. Today the situation is changed. User management pressures users to automate to control user organization costs. Requirements which in the past were more stable are so no longer. In many areas there are substantial changes each year that result in major enhancements and retrofitting.

o Problem Areas

Each of the surveys provided respondents with extensive lists of potential problem areas. The respondents were asked to rank these on a scale of 1 (no problem) to 5 (major problem). A variety of problem areas were listed and are summarized in Figure 1. Statistical analysis uncovered five main groupings of problem areas:

- User knowledge
- Programmer effectiveness
- Product quality
- Programmer time availability
- Machine requirements
- System reliability

Additional statistical factor analysis was performed to determine which factors contributed to the variance. The ranking was as that given above with user

FIGURE 1: POTENTIAL PROBLEM FACTORS IN MAINTENANCE SURVEYS

- o Maintenance personnel turnover
- o Documentation quality
- o System hardware and software changes
- o Demand for enhancements and extensions
- o Skills of maintenance programmers
- o Quality of original programming
- o Number of maintenance programmers available
- o Competing demands for programmer time
- o Lack of user interest
- o System run failures
- o Lack of user understanding
- o Program storage requirements
- o Program processing time requirements
- o Maintenance programmer motivation
- o Forecasting maintenance programming requirements
- o Maintenance programming productivity
- o System hardware and software reliability
- o Data integrity
- o Unrealistic user expectations
- o Adherence to programming expectations
- o Management support
- o Adequacy of system design specifications
- o Budgetary pressures
- o Meeting scheduled commitments
- o Inadequate user training
- o Turnover in user organizations

knowledge as the major component at 59.5% followed by programmer effectiveness at 11.9%. User knowledge includes user training and user expectation for changes as well as a lack of user understanding. Programming effectiveness includes skills of programmers as well as their productivity. In the surveys several potential problem areas which have been widely mentioned as concerns in the literature failed to be significant. These included processing and storage requirements, data integrity and hardware/software reliability. Migration across to new generations of hardware was not viewed as significant since the manufacturers provide software to aid such migration.

Personnel turnover impact was viewed as significant. This is due to the correlation found between the experience and time spent with the application system being inversely related to the degree to which maintenance of the system was perceived to be a problem. Maintenance effort was also found to negatively correlate with the time spent with the system.

With these problem areas highlighted we can turn again to the productivity aids. Most of the aids that have been developed to date (e.g., structured programming, HIPO, structured walk-through, etc.) have little impact on the two major components of the factor analysis.

In this section we have considered some of the major issues and problem areas uncovered in the maintenance surveys. In the next section we attempt to integrate these findings into a potential framework for maintenance.

3. TOWARD A FRAMEWORK FOR SOFTWARE MAINTENANCE

The issues presented in the previous section can be synthesized into a framework that may point to avenues of future work. This framework must include users, environment, systems groups, as well as the portfolio of applications being operated and maintained. Note that the attention is on the portfolio and not on a single application. Focusing on one application does not admit interproject interaction. Nor does it allow for resource sharing across applications. In the issues we noted that major issues relate to these more global concerns.

Considering a wider framework is also supported in the technology advances. As more complex technology is developed, the learning curve increases. In the past resources (personnel, hardware, and communications) could be devoted to single systems. This luxury is rapidly no longer available. Users reject high costs due to dedicated resources. Systems personnel turnover is higher. The number of people with high level skills is reduced compared to demand.

To build the framework we need to return to the categories of maintenance and individual activities. Each activity can be listed along with the responsibility of performing the activity. Figure 2 gives an example of such a list. It is not meant to be comprehensive or complete. It is merely an example of a categorization. Obviously, Figure 2 does not fit all environments. It is likely to be an optimistic view of what users can and are willing to do. Based on the data in the surveys this assignment would reallocate approximately 20% of the effort from the systems organization to the user organizations.

FIGURE 2: POSSIBLE CLASSIFICATION OF MAINTENANCE ACTIVITIES

Operations

- o Data entry- user
- o Inquiry- user
- o Production initiation- users (on-line systems) or operations (batch)
- o Fixing production problems- systems

Enhancements

- o Report generation from output files- users
- o Addition of new data elements- systems
- o Addition of new functions in system- systems
- o Modification of reports- users (if report generator is available)
- o Modification of tables to support system- user
- o Requirements analysis- user
- o Design- user, systems

Maintenance

- o Recoding for efficiency- systems
- o Improved documentation- systems support
- o Accommodation to changes in hardware, system software- systems
- o Accommodations to changes in files, input- systems

Management

- o Monitoring of change requests- user, systems
- o Project control- systems
- o Cost accounting- user, systems

After categorizing the activities the next step is to analyze the support requirements for hardware, software, and data communications. For the users to assume the roles in Figure 2 substantial easy to use software will be needed to perform data entry, report generation, inquiry, and analysis. Some of these tools exist, but they are not packaged well. The interfaces between the user and the systems are complex and even unique by system. Thus, the compelling need is for a standard user interface and support structure across applications. For example report generators must be general purpose across multiple accounting systems for an accounting organization.

There are implications for the systems organizations as well. Because of the increased user responsibilities a user support group must be in place. Such a support group would be able to train users in the use of tools. But this is not the only support needed. There must also be a group which is knowledgeable of the user environment and the data in the various files and data bases. This group will be referred to as the information support group. It would include the functions that many consider to be in the realm of decision support systems. The increased user role would likely result in savings which would more than compensate for these two groups. The application maintenance and enhancement activities now become more limited to major changes and tasks as opposed to relatively minor work involved in report generation.

Even with this restructuring there are still complications. The support for the users must cross applications. Furthermore, the hardware, system software, and data communications must also span the user organizations. Thus, the traditional system programming role must be generalized to include system software and the network of users.

Returning to Figure 2 again we can address the management activities. These are listed as being largely shared between users and systems managers. Project management becomes a dual responsibility. With the generality of support across applications it is necessary for a higher level of coordination to be in place to handle network wide issues and problems.

There are a number of implications that can be drawn from this new sharing of organizational responsibilities. We can first note that there are a number of productivity aids needed for users. Some of these in terms of ease of access are more available on microcomputers than large mainframes. A support structure of tools is needed to support the users' local computing needs.

Another area where work is needed is in the migration from the current systems environment to the new setting. This is not likely to be easy given the state of the current operational systems. There will be an unwillingness to expend the funds for major reworking of systems. This is an area where software aids would be useful in working with current applications.

A third implication concerns the management support structure. There must be tools to support single and multiple project management. These tools are not simple project control, PERT, or CPM programs. Rather they are interactive aids to work with managers.

On a larger management scale we must control change requests across the portfolio. To pursue this further consider the planning process. Assuming that a long range plan has been developed, there are strategies and project candidates which will aid the systems organization's long term objectives. These might include the use of new techniques and tools. At a certain point a slate of possible projects is created. This slate consists of projects from the following areas: 1) continuing maintenance and enhancement work, 2) large scale enhancements and development underway, 3) the backlog of unfilled requests for service, 4) long range planning candidates, 5) emergency, unpredictable requests, and 6) targets of opportunity. Given a portfolio management approach systems and user management must more proactively control the allocation of resources to these six areas. Without a formal allocation the activities that tend to fill the gaps are categories 1, 2, and 5. The backlog then continues to grow. The situation does not substantially change since the long range plan projects are left unfunded. The mixture of the six categories is dependent upon the organization, but the portfolio of projects actually approved would include projects from all categories.

Assuming that the framework is adopted for operational systems, there are inevitable effects on the development process and even on the approach to solving problems. In the current systems approach individual systems problems are addressed in small groups or individually by specific system solutions. As time passes, user organizations may have several separate and separately maintained systems. Overlaying these applications are new technologies (e.g., office automation).

Just as there is a need for an integrated framework in the systems area, there is the same need in the user organizations. Early stages of the life cycle (e.g., feasibility studies and user requirements) should be expanded to include the information and processing needs of the organization as a whole. System development then may involve coordination and installation of certain technologies simultaneous with traditional application development.

The surveys have revealed interesting specific results to date in technical and managerial areas such as productivity aids, controls, measurement, and organization. However, their most valuable long term contribution may be in the emphasis of an overall, more systematic structure for maintenance, enhancement, and operations.

REFERENCES

1. Belady, L.A. and M.M. Lehman. "A Model of Large Program Development." IBM Systems Journal 3, 1976, 225-252.
2. Boehm, B.W. "The High Cost of Software." Proceedings of Symposium on the High Cost of Software, Monterey, California, 1973, 27-40.
3. Boehm, B.W., J.R. Brown, and M. Lipow. "Quantitative Evaluation of Software Quality." Proceedings, Second International Conference on Software Engineering, San Francisco, October, 1976, 592-605.
4. Canning, R.G. (ed.) "That Maintenance Iceberg." EDP Analyzer, 17, January, 1979.
5. Lientz, B.P. and E.B. Swanson. "Discovering Issues in Application Software Maintenance." Data Management, 16, September 1978.
6. Lientz, B.P. and E.B. Swanson. "Software Maintenance: A User/Management Tug-of War." Data Management, 17, April 1979, 26-30.
7. Lientz, B.P. and E.B. Swanson. Software Maintenance Management. Reading, Mass.: Addison-Wesley Publishing Co., 1980.
8. Lientz, B.P., E.B. Swanson, and G.E. Tompkins. "Characteristics of Application Software Maintenance." Communications of the ACM. 21, 1978, 466-471.
9. Swanson, E.B. "The Dimensions of Maintenance." Proceedings, Second International Conference on Software Engineering. San Francisco, October, 1976, 492-497.

EN
DATE
ILME